LEGIBILITY NOTICE

A major purpose of the Technical Information Center is to provide the broadest dissemination possible of information contained in DOE's Research and Development Reports to business, industry, the academic community, and federal, state and local governments.

Although a small portion of this report is not reproducible, it is being made available to expedite the availability of information on the research discussed herein.

CONF-881223--1

LA-UR 88-1792

LA-UR--88-1792

DE88 014466

Received by OSTI

AUG 04 1988

TITLE: A DOCUMENT PREPARATION SYSTEM IN A LARGE NETWORK ENVIRONMENT

AUTHOR(S): Manuel Vigil, C-6
Stephany Bouchier, C-6
Claudia Sanders, C-2
Steve Sydoriak, C-2
Kyle Wheeler, C-2

SUBMITTED TO: ACM Conference on Document Processing Systems
December 5-9, 1988
Santa Fe, NM

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

A DOCUMENT PREPARATION SYSTEM IN A LARGE NETWORK ENVIRONMENT

by

Manuel Vigil and Stephany Bouchier
Computer Graphics Group
Claudia Sanders, Steve Sydoriak, and Kyle Wheeler
Computer Documentation Group
Computing and Communications Division
Los Alamos National Laboratory
Los Alamos, NM 87545

ABSTRACT

At Los Alamos National Laboratory, we have developed an integrated document preparation system that produces publication-quality documents. This system combines text formatters and computer graphics capabilities that have been adapted to meet the needs of users in a large scientific research laboratory. This paper describes the integration of document processing technology to develop a system architecture, based on a page description language, to provide network-wide capabilities in a distributed computing environment. We describe the Laboratory requirements, the integration and implementation issues, and the challenges we faced developing this system.

INTRODUCTION

Systems for document preparation have been in use at the Laboratory for many years. These systems have traditionally been tightly coupled with specific computing environments and output devices. There was an enormous cost associated with the use and support of all these incompatible systems. Recognizing this, the Publication Strategy Team (PST) was formed to identify directions for document processing at the Laboratory. The team's initial recommendations provided the early direction for integrating document processing systems. In the years since the team was formed, there have been many advances in technology for document processing systems. These rapid changes have provided us with additional opportunities and challenges as we have planned and set up an integrated system that is continually evolving.

This paper describes the project to develop our document preparation system. We describe the project background, network integration issues, text formatting systems, and merging text and graphics. We also describe support for this system and describe some future plans.

PROJECT BACKGROUND

Details about our environment, our vision, and our goals will aid the understanding of the design and implementation of our document preparation system.

The Los Alamos Environment

The Laboratory has one of the most powerful scientific computing centers in the world, with resources that range from powerful supercomputers to scientific workstations and personal computers. Major computing resources are located in the Integrated Computing Network (ICN), which includes numerous computers manufactured by Cray Research, Digital Equipment, IBM, and Control Data; a central file storage system, called CFS, allows users to store and retrieve their files from any host computer; and a central output node, called PAGES, generates print and graphics output (see Figure X). Dependence on a central output station was ingrained in our users' minds and plays an important part in the architecture developed for our document preparation system.

Users of document processing systems at Los Alamos include scientists, technical writers, editors, word processors, illustrators, and secretaries. The types of documents they produce, which range in size from one page to several hundred pages, include technical reports, scientific publications, memos, letters, and computer documentation. The applications they use to produce documents reside on mainframes, workstations, and personal computers.

Vision

We envision an environment where different users can generate different types of documents using an application on the most appropriate computing system. This vision includes support of tools on each computing environment so that we can maximize the productivity of users. We also want to make it easy to share data between applications and computing systems. For example, a person using a text formatter on a mainframe computer should have the capability to electronically incorporate computer graphics created on that same computer system or graphics generated or scanned on a desktop system. Conversely, a person using a desktop publishing system should also be able to include graphics created either place.

With the recent craze in desktop publishing why don't we just use desktop systems to solve all of our document preparation needs? While desktop publishing applications are excellent for some types of documents, they are still not capable of doing it all. Additionally, the reality of the situation is that there are hundreds of documents at the Laboratory that have been prepared using text formatting systems; the

source files for these documents are stored for later updates and revisions. The cost of converting these documents to new systems is prohibitive. While the trend at the Laboratory is to use new systems where applicable, the existing systems have so much invested in them that they require support for the near future. The goal is to make the existing applications fit into a system foundation that allows users to easily convert to new systems when appropriate.

Project Goals

The original goals for the implementation of the document preparation system addressed the needs of the users as well as the needs of the system implementors. These goals were defined in the report prepared by the PST.(Reference) For the users, goals were directed at providing as much functionality and flexibility as possible in the task of document preparation. For users the system had to be

- Capable of producing any type of document.
- Easy to use and understand by technical and secretarial staff.
- Capable of handling large, long-lived technical documentation, with independent formatting of parts.
- Capable of merging text and graphics.
- Capable of handling classified information.
- Capable of printing on a variety of output devices.
- Extensible for individual tailoring.

From an overall system point of view the integrated system had to

- Support a variety of text formatters.
- Support tools for merging text and graphics.
- Support both local and distributed output (on the central output node).
- Be integrated into the existing computer network and be accessible from several operating systems.
- Support medium- and high-quality output.
- Use state-of-the-art technology.
- Use standards as much as possible.
- Be able to evolve, providing facilities to meet changing needs.

The project team was formed in 1984 and consisted of people with text formatting experience, computer graphics experience, and network integration experience. Even though the project had many goals, the major goals initially were to provide a system foundation for merging text and graphics and for generating identical output on printers in users' offices and on PAGES. Thus, the project's emphasis was on three areas: network integration for output on PAGES, text formatting capabilities, and merging of text and graphics. The following sections describe the major issues, challenges, and solutions in each of these areas.

NETWORK INTEGRATION

The three major areas of emphasis in implementing a document preparation system at the Laboratory were text formatting, graphics generation, and network integration. Initially, the three were highly intertwined. The existing graphics system was already well integrated in the network. We used our experience implementing the graphics system to guide the design of the document preparation system.

Major Issues

We knew we could use existing formatting systems to satisfy users' requirements for our document preparation system, but the requirements for network integration actually shaped the system architecture. The important questions we had about network integration included:

1. Where should we provide formatting capabilities and output?

2. What type of file should go through the network: source file for the formatter or an intermediate file created by the formatter?

3. How should formatted text be merged with the locally produced graphics metafile?

4. How can we provide identical output on all printers, local and central, medium and high quality?

5. What should the user interface be?

We conducted a feasibility study to see if formatted text output could be integrated into the graphics system. Figure Y shows the data path from a TROFF system to the graphics metafile. Using the graphics system would have provided solutions to most of the major questions identified above. Unfortunately, when graphics stroke fonts were substituted for the original TROFF fonts, intercharacter-spacing problems made the system unacceptable for publication-quality documents.

At about this time the concept of a page description language was being introduced with practical applications. Both Adobe Systems Incorporated and Xerox Corporation announced the implementations of page description languages (Adobe was developing PostScript, and Xerox was developing Interpress). Both provided features that could be useful to us: device-independent representation of both textual and graphical information; unlimited functionality through the use of a language for creating shapes and transforming them in various ways; and a possible de facto standard for printer output. Information on

PostScript was generally available, but it was more difficult to identify the status of Interpress. It also became apparent to us that there were more printer manufacturers and software companies implementing PostScript than Interpress. Comparisons of the two (reference) helped us select PostScript as the standard page description language at the Laboratory. PostScript was more flexible and didn't have many implementation subsets, as did Interpress.

After prototyping several alternatives, we decided on an implementation scheme. We defined PostScript as a local "standard document file" for network transmission. Host machines translate data both from text formatters and from graphics into PostScript, merging all data for an individual document into one large file. The file can then be output on a local printer or shipped through the network to PAGES.

There are two advantages to this approach: we create a single file that can be output identically on either local or central printers, and the transmission of a single file through the network, in spite of its large size, avoids the possible loss of any separate part of a document job.

Integrating PostScript on PAGES

The use of PostScript as the "standard document file" at the Laboratory provided some interesting challenges. Using PostScript on PAGES caused some concern. Since then, many of those concerns have been allayed both by progress in the industry in adopting PostScript as a de facto standard and by our increased experience with PostScript.

Concerns about using PostScript on PAGES were based on the fact that PostScript is not only a page description language but also a general-purpose language. Until PostScript came along, PAGES evaluated all incoming data before sending it to output devices, throwing away bad data to keep the device from wasting time on it. The difference between PostScript and data previously accepted by PAGES was that, because PostScript is a language, it could control a device, as with an infinite loop, thereby shutting down network printing services. Because of our work with classified data, the idea that PostScript could take over a device was a serious concern.

The tools we've supported to generate PostScript have been responsible and have reduced our concern about bad PostScript data. However, we are still faced with the specter of illegitimate PostScript programs that misuse the device and the attendant security and misuse implications. We have also found that there are ways (not always simple) to preprocess the data and even interpret the PostScript on a host computer, where we have better control of it. We also have had to accept that existing interpreters are very slow.

although we believe that that situation will improve as more printer manufacturers adopt PostScript.

PostScript is now very much a part of the central output node. We currently support PostScript output on 8-1/2- X 11-inch 300-dpi paper. Eventually, we expect to support high-resolution paper, film, and possibly color paper devices.

The PostScript Environment in the Network

We have found PostScript to be very useful in our distributed network enviroment. The functional richness it offers overcomes its relative slowness, at least for now. Certainly there are printers that can do things faster, or fonts that take up less space, or more efficient data formats for page description languages, but there is something to be said for something that works well, works every time, and can process all the information it is given.

The use of PostScript as a standard in the network has had a tremendous influence on the productivity of users and systems people. PostScript is the only format supported for graphics output on local printers. This saves much time formerly spent writing device drivers. TeX and TROFF both generate PostScript. Since they are the two text formatters that we support, we no longer need intimate knowledge of every printer, as we did before in supporting text formatters. The great majority of document production is now handled by PostScript. In addition, by basing our system on PostScript, we are able to include PostScript files from the Macintosh environment with formatted text from our network's mainframes.

TEXT FORMATTING SYSTEMS

When we first surveyed document preparation systems in 1984, electronic publishing was just beginning to attract interest in the global computing community. Since one of the earliest uses of the UNIX operating system had been for text processing, TROFF was a primary candidate for a text formatter in our document preparation system. Besides its text processing features, TROFF has preprocessors for math, tables, and elementary graphics. Some standard macro packages were available, and locally written macro packages supported Laboratory standards for memos, letters, and computer documentation. The macro packages had the added bonus of providing a declarative language interface for the user. In addition, a base of expertise in the TROFF system existed at the Laboratory because TROFF had been used for preparing documents since 1976.

Based on these considerations, TROFF was selected for most of the initial work toward the implementation of the document preparation system. Conveniently, Adobe Systems Incorporated provided the

Transcript filters from TROFF to PostScript early on. With these filters and an early PostScript printer, we were able to demonstrate the feasibility of our network documentation system in the spring of 1986.

Accomplishing our original goals would be difficult if TROFF were the only text formatter available. One major problem was that TROFF was only available on the UNIX operating system, so there was no way to use it to produce classified documents. While there was enough use of TROFF to continue support, there was also reason to support another text formatter for the user community.

We decided to also support the TeX text formatting system. TeX is available on all major systems--UNIX and VMS on VAX computers, IBM Personal Computers, Sun and Apollo workstations, and Macintosh personal computers. We are also porting TeX to CTSS (the Cray Timesharing System) on the Cray computers. TeX produces high-quality documents for a wide variety of situations, and it does an excellent job of formatting mathematics, which is a big plus in our scientific and mathematical community. The breakthrough that enabled us to support TeX in our PostScript-based environment was the release of the DVIPS software, which translates the device-independent (DVI) file produced by TeX into PostScript. DVIPS comes from ArborText.

One of the problems with TeX is that it is difficult to learn and use. However, macro packages such as LaTeX can reduce that concern. LaTeX is easy to use and has a declarative interface, which frees the user from most formatting decisions. We have provided users with LaTeX macros to print Laboratory-standard memos, and we plan to provide macros to print Laboratory-standard letters and reports.

Handling of fonts was easy with TROFF because all fonts convert to use the outline fonts in PostScript. TeX, on the other hand, was more difficult because of the multisystem environment and because the standard PostScript fonts did not include all those needed by TeX. We considered various ways to handle fonts. It is possible to download fonts into a PostScript printer, but PostScript's memory is only big enough to hold three of TeX's 50-some commonly used fonts. The other possibility is to send TeX's DVI file and any PostScript graphics files over to PAGES and to run DVIPS on PAGES. All the TeX fonts would then be stored on a PAGES host computer, and the DVI file would only carry the font name over the network. We decided to run DVIPS on the local machine and ship the more bulky PostScript file over the network. It is more reliable to send a single PostScript file to PAGES than to send a DVI file and a possibly large number of graphics files.

We would like to integrate a WYSIWYG formatter in the document preparation system. Many people like WYSIWYG formatters because they are easy to use. Unfortunately, we have not yet found one that has a sufficiently transportable source file. WYSIWYG document files are usually loaded with non-ASCII characters that make them difficult to read on another type of computer. It is even more difficult to edit text in a WYSIWYG file from another computer. It is usually downright impossible to edit tables or equations in a WYSIWYG file from another computer. Most WYSIWYG formatters do not do a good job with math. Some formatters require that you position math components by hand. Others produce equations that are not professional in appearance.

Eventually we hope to find a WYSIWYG formatter that will have a transportable document source file. The most likely design for such a product at this time would be a WYSIWYG formatter that could produce a TeX source file. This would enable the user to easily prepare a document on the WYSIWYG system, while allowing the document to be circulated to other systems for editing and printing.

MERGING TEXT AND GRAPHICS

The computer graphics environment at the Laboratory is system and device independent. Users have access to the same graphics capabilities on all the major systems (CTSS on the Cray computers and VMS and UNIX on the VAX computers). The Common Graphics System (CGS) uses the concept of a metafile, a device- and system-independent data file, which is used for previewing output on the screen and for generating graphical output on PAGES.

The original goal of merging text and graphics was well focused. The intent was to merge graphics from the CGS metafile with formatted text. Since all computer graphics software produced the metafile, we had to write only one interface to translate between graphics data and PostScript. Since then, other graphics capabilities have been introduced, which users also want to include in their output. The remainder of this section describes the graphics issues, merging graphics with the TROFF and TeX formatters, and some future plans in the graphics area.

Metafile-to-PostScript Translation

PPS is the utility we wrote to translate graphics from our CGS metafile into PostScript. This tool accepts user options specifying output size and placement of the resulting image on the page. It provides such things as an overlay feature for combining two images on the same page. PPS is widely used at the Laboratory for independent graphics output as well as for graphics to be included in a document.

Although PPS has provided a much needed capability, we recognize the need for an additional, more efficient graphics package based on PostScript. The PostScript files produced by PPS are inefficient compared to the original metafile. The most inefficient feature is that any text included in graphics ends up in stroked software fonts by the time it ends up in the metafile, so, of course, text in the PostScript files is made up of many moves and draws. A package that directly produces PostScript could take advantage of the PostScript fonts. Such a package could also provide other features, such as control of line width and line style. For these reasons, we are beginning to design a graphics package to take direct advantage of PostScript capabilities.

TROFF Integration

We have provided a TROFF macro to include graphics in TROFF output. The macro uses either a CGS metafile or a PostScript file. When a metafile is used, the macro calls the PPS utility to translate the metafile into PostScript. The macro has arguments to provide flexibility in placing the figure in a document. Figure 2 shows what happens when the macro is called. We will continue to expand this capability to include graphics from other environments into TROFF documents.

TeX Integration

We are using vendor printer drivers or public domain software to integrate graphics into TeX. With DVIPS a user can include the \SPECIAL command to integrate PostScript files. The \SPECIAL command allows the inclusion of only PostScript, so a user must first run PPS to translate a CGS metafile or produce PostScript files some other way.

USER SUPPORT

Peripheral C-Division support of the document preparation system contributes to its success. This support is provided in the forms of site licenses, author tools, documentation, education, and consulting. More details are given below.

Site licenses

Besides saving money and time, site licenses ensure the use of the same versions of software across the systems. This gives us confidence that tools we base on vendor software will be consistent on all systems. It also allows us to update software as new versions become available.

From ArborText, we bought site licenses for TeX products on all our major systems, including VAX/UNIX, VAX/VMS, Sun and Apollo workstations, and IBM PCs. TeX products include TeX, printer drivers for PostScript (DVIPS) and LaserGrafix printers, and previewing software. We also bought a site license for the fonts that are used on our standard memo and letter paper.

We purchased site licenses for Documentors Workbench (DWB) and Writers Workbench (WWB) from AT&T. DWB includes device-independent TROFF with standard macro packages and the preprocessors. WWB contains a large set of author tools including spelling and grammar checkers.

Author Tools

Locally written author tools complement the existing system and provide features specific to the needs of Los Alamos users. One of the most useful tools is a utility written for automatic index generation of TROFF documents. The index program sorts index entries flagged by the author and inserts format commands to create a final index. A macro package complements the index program by providing the author some flexibility in format details.

Documentation, Education, and Consulting

Other areas of concern are documentation of systems, user education, and consulting. The Computing Information Center at the Laboratory distributes vendor documentation and locally produced documentation that describes the various components of the document preparation system.

As a further help to the user community, videotapes featuring some of our system capabilities were produced and broadcast on LABNET, the Laboratory's educational television network. The tapes and student materials are available at the Computing Learning Center.

Consulting is available to users of the ICN during normal working hours.

FUTURES

The use of PostScript as the page description language gives us a very flexible foundation for the future. The major future directions are in enhancing the document preparation system we have developed to include capabilities from the desktop environment, include scanned or synthetic images, and enhance the current computer graphics environment to produce true publication-quality graphics. We will also provide ways to generate high-resolution output as well as color output from this system.

The desktop environment, especially the Macintosh, supports PostScript as well as other file formats. As mentioned before, this support of PostScript allows us to take graphics generated on the Macintosh and incorporate them in TeX or TROFF files. The tools available on the Macintosh provide an ideal solution for scanning and editing grey-scale images. They can then be incorporated in a page layout program or sent to another operating environment to be included in a text formatting system. We plan to continue work in this area.

Now that users have seen the capabilities that are possible with our system, they are demanding higher-quality graphics output. This can be done by using scientific workstation applications or desktop applications. However, many of these users want to generate their publication-quality graphics on the supercomputers where all of their data or applications reside. As mentioned before, a graphics package will be developed or modified to produce publication-quality graphics as defined by the Laboratory. We will work with graphics illustrators and users to develop this package. This application will be enhanced to operate in a distributed window system environment to take advantage of the computational power of the supercomputers and the interactive graphical interfaces of the workstations.

CONCLUSION

To be written later.

ACKNOWLEDGMENTS

We thank Steve Smith, David Kratzer, and Ernest Buenafe for their contributions. We also thank Ann Solem and the other members of the Publication Strategy Team.

This work was performed under the auspices of the U.S. Department of Energy under contract W-7405-Eng.36.

REFERENCES

To be added later.